CMPT 280

Topic 1: Review of Lists

Mark G. Eramian

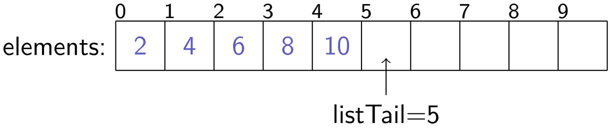
University of Saskatchewan

References

- Textbook, Chapter 1

Array-based List Implementations

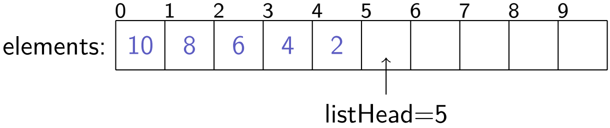
Implementation 1: Ordered first-to-last at beginning of the array



	0	1	2	3	4	5	6	7	8	9
elements:	2	4	6	8	10					

listTail=5

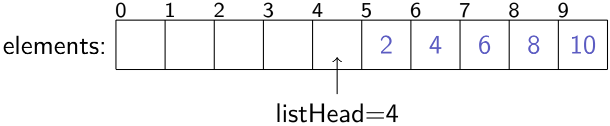
Implementation 2: Ordered last-to-first at beginning of the array



	0	1	2	3	4	5	6	7	8	9
elements:	10	8	6	4	2					

listHead=5

Implementation 3: Ordered first to last at end of array



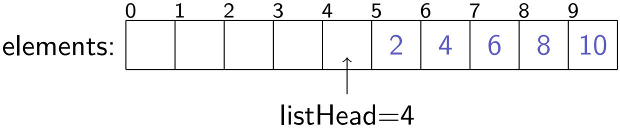
	0	1	2	3	4	5	6	7	8	9
elements:						2	4	6	8	10

listHead=4

Exercise 1

- Write a java class definition for the list implementation 3 (from the reading). Define the instance variables and a constructor, but for now, don't worry about defining the methods. Can you write it so we can store any type of element we want in the list?

Implementation 3: Ordered first to last at end of array



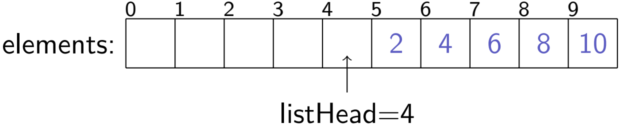
	0	1	2	3	4	5	6	7	8	9
elements:						2	4	6	8	10

listHead=4

Exercise 2

- Write two methods that test whether the list is full and empty, respectively.

Implementation 3: Ordered first to last at end of array



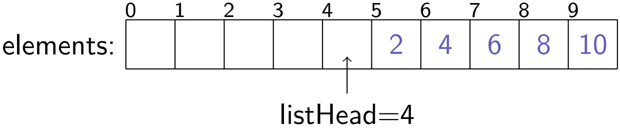
	0	1	2	3	4	5	6	7	8	9
elements:						2	4	6	8	10

listHead=4

Exercise 3

- Write the `insertFirst` method for our array-based list class which inserts a new element at the beginning of the list.

Implementation 3: Ordered first to last at end of array



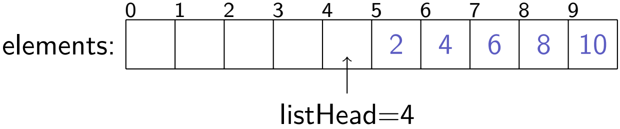
	0	1	2	3	4	5	6	7	8	9
elements:						2	4	6	8	10

listHead=4

Exercise 4

- Write the `deleteFirst` method for our array-based list class that removes the element at the beginning of the list.

Implementation 3: Ordered first to last at end of array



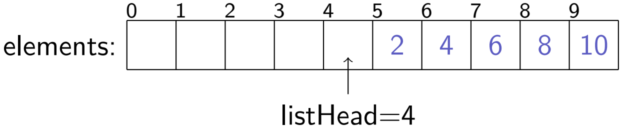
	0	1	2	3	4	5	6	7	8	9
elements:						2	4	6	8	10

listHead=4

Exercise 5

- Write the `firstItem` method for our array-based list class that returns the data element at the beginning of the list, but does not modify the list.

Implementation 3: Ordered first to last at end of array

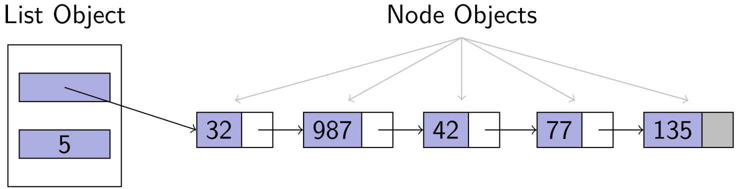


	0	1	2	3	4	5	6	7	8	9
elements:						2	4	6	8	10

listHead=4

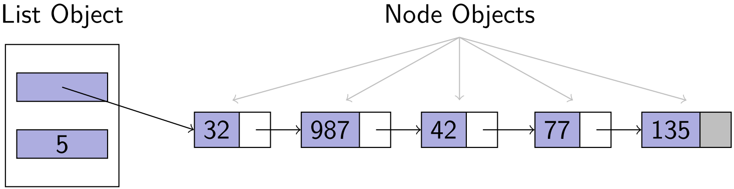
Linked Lists

Recall the structure of a singly-linked list:



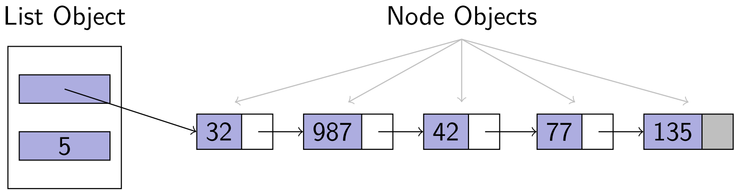
Exercise 6

- Write the class definitions (instance variables and constructors only) for the Node and List objects. Again, we would like to be able to store elements of any type.



Exercise 7

- Write the following methods for the list:
    - `isEmpty`
    - `insertFirst`
    - `deleteFirst`
    - `firstItem`



Observations

- Both versions of our list look the same to the user – same interface, different internals!
- Ok, fine... same interface except for `isFull()`. But we can easily fix that. How?
- How do we know whether what we just wrote works?
- Next class reading: Chapter 2 – Regression Testing.